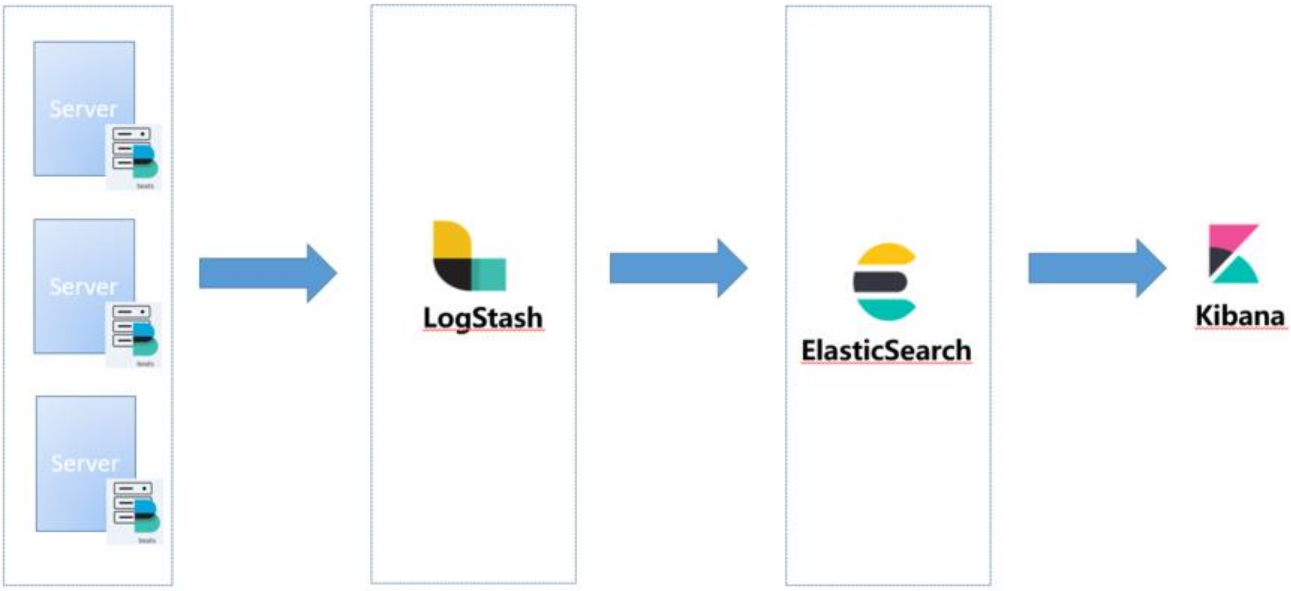
EFK 实战一 - 基础环境搭建

作者：jianzh5
原文链接：https://ld246.com/article/1578302649788
来源网站：链滴

在分布式系统中，由于节点服务会部署多台，一旦出现线上问题需要通过日志分析定位问题就需要登服务器一台一台进行日志检索，非常不便利，这时候就需要用到EFK日志收集工具。

在应用服务端部署Filebeat，将我们打印到日志文件中的日志发送到Logstash中，在经过Logstash的析格式化后将日志发送到ElasticSearch中，最后通过Kibana展现出来。EFK基础版的架构如下：



本文主要是使用docker和docker-Compose部署ELK的基础环境，选择7.5.1作为EFK组件版本。

当然了如果大家对docker，docker-compose不是很熟悉的话可以翻看我之前为大家准备的两篇文章：

- Docker基础与实战，看这一篇就够了
- Docker-Compose基础与实战，看这一篇就够了

实在不想使用docker部署的话也可以下载对应的安装包然后手动部署，配置方式基本一样。

安装配置

elasticsearch

安装elasticsearch之前先配置如下的系统变量

- 修改 /etc/sysctl.conf，在最后追加如下配置

```
vm.max_map_count = 655360
```

- 修改 /etc/security/limits.conf，增加如下配置

```
*          soft    memlock         unlimited
*          hard    memlock         unlimited
*          hard    nofile          65536
*          soft    nofile          65536
```

- 修改 /etc/security/limits.d/20-nproc.conf，增加如下配置

```
*        soft    nproc     4096
root      soft    nproc     unlimited
```

- 启动elasticsearch临时容器

```
docker run --rm --name es -p9200:9200 -p9300:9300 -e discovery.type=single-node elasticsearch:7.5.1
```

- 导出elasticsearch配置文件

```
docker cp fbce586c8a56:/usr/share/elasticsearch/config/elasticsearch.yml /app/elk/elasticsearch/conf/elasticsearch.yml
```

- 修改es配置文件

```
cluster.name: "elk-cluster"
network.host: 0.0.0.0
bootstrap.memory_lock: true
discovery.type: single-node
```

- 建立es的日志文件夹和数据文件夹，并对文件夹授权

```
mkdir -p /app/elk/elasticsearch/logs
mkdir -p /app/elk/elasticsearch/data
```

```
chmod -R 777 /app/elk/elasticsearch/logs
chmod -R 777 /app/elk/elasticsearch/data
```

- 停止临时容器

```
docker stop fbce586c8a56
```

logstash

- 启动临时容器

```
docker run --rm --name logstash -p5044:5044 -p9600:9600 logstash:7.5.1
```

- 导出docker的配置文件

```
docker cp 5adb0971bb0f:/usr/share/logstash/config /app/elk/logstash
```

- 建立logstash数据文件夹，并对其授权

```
mkdir -p /app/elk/logstash/data
chmod -R 777 /app/elk/logstash/data
```

- 复制logstash启动文件，并对其修改

```
cd /app/elk/logstash/config
cp logstash-sample.conf logstash.conf
```

修改logstash.conf，配置output

```
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["http://172.31.0.207:9200"]
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    #user => "elastic"
    #password => "changeme"
  }
}
```

暂时修改一下ES的访问路径即可。

- 停止临时容器

```
docker stop 5adb0971bb0f
```

kibana

- 启动临时容器

```
docker run --rm --name kibana -p5601:5601 kibana:7.5.1
```

- 导出kibana配置文件

```
docker cp f21f0f9e0259:/usr/share/kibana/config/kibana.yml /app/elk/kibana/conf/kibana.yml
```

- 修改kibana配置

```
server.name: kibana
server.host: "0"
elasticsearch.hosts: [ "http://172.31.0.207:9200" ]
xpack.monitoring.ui.container.elasticsearch.enabled: true
i18n.locale: zh-CN
```

设置i18n.locale: zh-CN属性后会对kibana进行汉化，这样便于操作，主要还是我英语不太好~

- 停止临时容器

```
docker stop f21f0f9e0259
```

docker-compose

经过上面的准备，我们可以编写docker-compose文件，方便我们对容器进行编排，一键启动。有了前的基础，我们很容易编写出对应的yml文件，编写后的内容如下：

```
version: "3"
services:
  elasticsearch:
    image: docker.io/elasticsearch:7.5.1
    container_name: elasticsearch
    environment:
      - "ES_JAVA_OPTS=-Xms4096m -Xmx4096m -Xmn1300m"
    volumes:
      - /app/elk/elasticsearch/conf/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsear
h.yml
      - /app/elk/elasticsearch/data:/usr/share/elasticsearch/data:rw
      - /app/elk/elasticsearch/logs:/usr/share/elasticsearch/logs:rw
    ports:
      - "9200:9200"
      - "9300:9300"
    restart: always

  kibana:
    image: docker.io/kibana:7.5.1
    container_name: kibana
    volumes:
      - /app/elk/kibana/conf/kibana.yml:/usr/share/kibana/config/kibana.yml
    ports:
      - "5601:5601"
    depends_on:
      - elasticsearch
    restart: always

  logstash:
    image: logstash:7.5.1
```

```
  container_name: logstash
  command: logstash -f /usr/share/logstash/config/logstash.conf
  volumes:
    - /app/elk/logstash/config:/usr/share/logstash/config
    - /app/elk/logstash/data:/usr/share/logstash/data
  ports:
    - "9600:9600"
    - "5044:5044"
  depends_on:
    - elasticsearch
  restart: always
```

将docker-compose文件上传至服务器，启动docker服务

docker-compose -f elk.yml up -d

```
[root@bingo-172 config]# docker ps
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS                                                  NAMES
c9159e64c697        logstash:7.5.1          "/usr/local/bin/dock…"   3 days ago          Up 3 days           0.0.0.0:5044->5044/tcp, 0.0.0.0:9600->9600/tcp         logstash
d9c962b7f5ff        kibana:7.5.1            "/usr/local/bin/dumb…"   3 days ago          Up 5 hours          0.0.0.0:5601->5601/tcp                                 kibana
2a912f85db9a        elasticsearch:7.5.1     "/usr/local/bin/dock…"   3 days ago          Up 5 hours          0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp         elasticsearch
```

启动完成后访问kibana地址http://172.31.0.207:5601/验证是否正常访问

安全认证

我们刚刚部署的elk环境是不需要密码就可以登录kibana的，这样谁都可以访问而且可以更改数据。以我们需要给kibana加个密码，必须要登录才可以进行操作。

主要是利用elasticsearch自带的xpack作为权限验证功能。操作步骤如下：

- 修改es外部配置文件 /app/elk/elasticsearch/conf/elasticsearch.yml，开启权限验证

```
xpack.security.enabled: true
```

- 重启 elasticsearch服务

docker-compose -f elk.yml restart elasticsearch

- 进入es容器，为内置账号设置密码

```
docker exec -it  elasticsearch /bin/bash
cd /usr/share/elasticsearch/bin
./elasticsearch-setup-passwords interactive
```

```
[root@2a912f85db9a bin]# ./elasticsearch-setup-passwords interactive
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,logstash_system,beats_system,remote_monitoring_user.
You will be prompted to enter passwords as the process progresses.
Please confirm that you would like to continue [y/N]y


Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana]:
Reenter password for [kibana]:
Passwords do not match.
Try again.
Enter password for [kibana]:
Reenter password for [kibana]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Passwords do not match.
Try again.
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Changed password for user [apm_system]
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
[root@2a912f85db9a bin]#
```

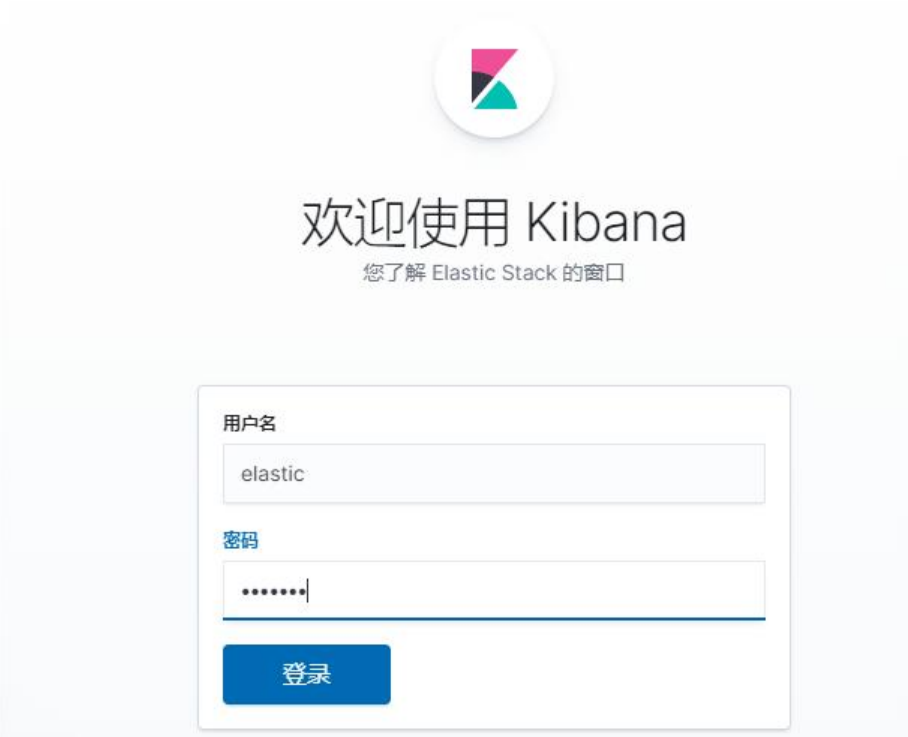
- 修改kibana配置文件 /app/elk/kibana/conf/kibana.yml

elasticsearch.username: "elastic"
elasticsearch.password: "xxxxxx"

- 重启kibana

docker-compose -f elk.yml restart kibana

- 重新访问kibana，并使用上面设置的elastic账号进行登录



至此我们顺利给ELK加上了安全认证，可以放心在生产环境部署使用了！

好了，各位朋友们，本期的内容到此就全部结束啦，下一期我们会将业务系统的日志接入ELK并对日

进行解析格式化，欢迎持续关注。

如果觉得这篇文章对你有所帮助的话请扫描下面二维码加个关注。"转发" 加 "在看"，养成好习惯！咱下期再见！

原文链接：EFK 实战一 - 基础环境搭建